BBC RD 1985/1

BBC

RESEARCH DEPARTMENT

REPORT

The 20:16 block code for digital recording and transmission channels

M.A. Parker, B.Sc.

Research Department, Engineering Division
THE BRITISH BROADCASTING CORPORATION

January, 1985

BBC RD 1985/1
UDC 621.397.61:
621.394.149

THE 20:16 BLOCK CODE FOR DIGITAL RECORDING AND TRANSMISSION CHANNELS

M.A. Parker, B.Sc.

Summary

This Report describes a new channel code which offers several advantages; these include a higher frequency of data transitions in the serial coded data, a greater ability to detect errors, and the inclusion of rugged framing information. It was devised for use with magnetic tape recording but is equally applicable to other forms of recording, and has been used for optical fibre transmission.

Issued under the Authority of

M. E. B. Moffat

Head of Research Department

Research Department, Engineering Division,
BRITISH BROADCASTING CORPORATION

January, 1985
(PH-261)

THE 20:16 BLOCK CODE FOR DIGITAL RECORDING AND TRANSMISSION CHANNELS

THE 20:16 BLOCK CODE FOR DIGITAL RECORDING AND TRANSMISSION CHANNELS

M.A. Parker, B.Sc.

1. Introduction

When recording digital signals on magnetic tape, channel coding is used to convert the input NRZ (non return to zero) data to a form that can be replayed more easily. Low frequency and d.c. signals are not normally recoverable during replay when using conventional heads sensitive to changes in magnetic flux, and so it is desirable to code a sequence of input bits to be recorded into a sequence of coded bits which possesses a nearly equal number of binary 'ones' and 'noughts'. Also, frequent data transitions must occur in the coded domain to allow accurate resampling of the recovered data. This also reduces crosstalk between adjacent tracks and reduces the depth of the recording into the tape allowing signals to be over recorded without the need to separately erase the tape[1].

The coding can either be effected by processing groups of input bits as independent blocks or by a continuous, recursive, bit by bit process which takes account of the sequence of bits already coded. The former normally allows the precise nature of the coded data to be controlled more accurately since the code words may be chosen freely and stored in a look-up table. Examples of this type of code include 3PM[2]; 10:8[3] and 8:14[4]. Recursive coding is more restrictive in this respect since suitable coding algorithms must be devised. Their advantage in the past has resulted from the low cost of implementation, although with the reduction in cost of semi-conductor memory, block coding has become comparable in this respect. Biphase[5] and delay modulation[6] are examples of this type of coding.

2. Choice of code

A block code can be characterised as a process in which the data to be coded is divided into blocks of n bits which are coded to a larger block of m bits. Of the total number of possible coded permutations of 2^m only 2^n are used. These can be selected to optimise the properties of the code. In general the longer the block of data to be coded, the more efficient the code. This is due to the need to match the bit patterns at the beginning and end of the codewords to ensure the desired coded data characteristics at the transition between adjacent words under 'worst case' conditions.

Various combinations of n and m were tested by computer to find a convenient codeset which would be d.c. free and have a limited maximum distance between transitions. A 20:16 code was found (m = 20, n = 16) which was particularly efficient, incurring a 25% increase in data rate and allowing a codeset slightly greater than 2^{16}.

Conventionally, a look-up table stored in PROM (Programmable Read Only Memory) would be used for the coding and decoding operations: the input data would be applied as the address to the semiconductor memory and the appropriate codeword would reside at that address. On decoding, the codewords would address another memory which would contain the full set of input data permutations. In the case of 20:16 code, the size of those tables would total about 16 Mbits which at present is prohibitive both in terms of size and cost. The storage requirements for coding and decoding have therefore been reduced by using a systematic process of coding which relies on using a small amount of storage (about 2 kbytes) and some sequential coding logic[7].

3. Implementation of the 20:16 code

3.1. General

The method of coding involves dividing the input data word into two smaller blocks. The first of the small blocks is coded as an entity; the coding of the second block is then carried out taking regard of any limitations imposed by the nature of the first.

Referring to Fig. 1, the sixteen bit input word is divided into two smaller blocks, one of nine bits and the other of seven. Each of these is transformed into ten bits of output data. The codeset is chosen so that each ten-bit output block has the following properties:

a) it is composed of either four '1's and six '0's (a 4:6 word), five '1's and five '0's (a 5:5 word) or six '1's and four '0's (6:4 word).

b) with reagard to digits at the beginning and end of each ten bit block, no more than the first three and no more than the last two may have the

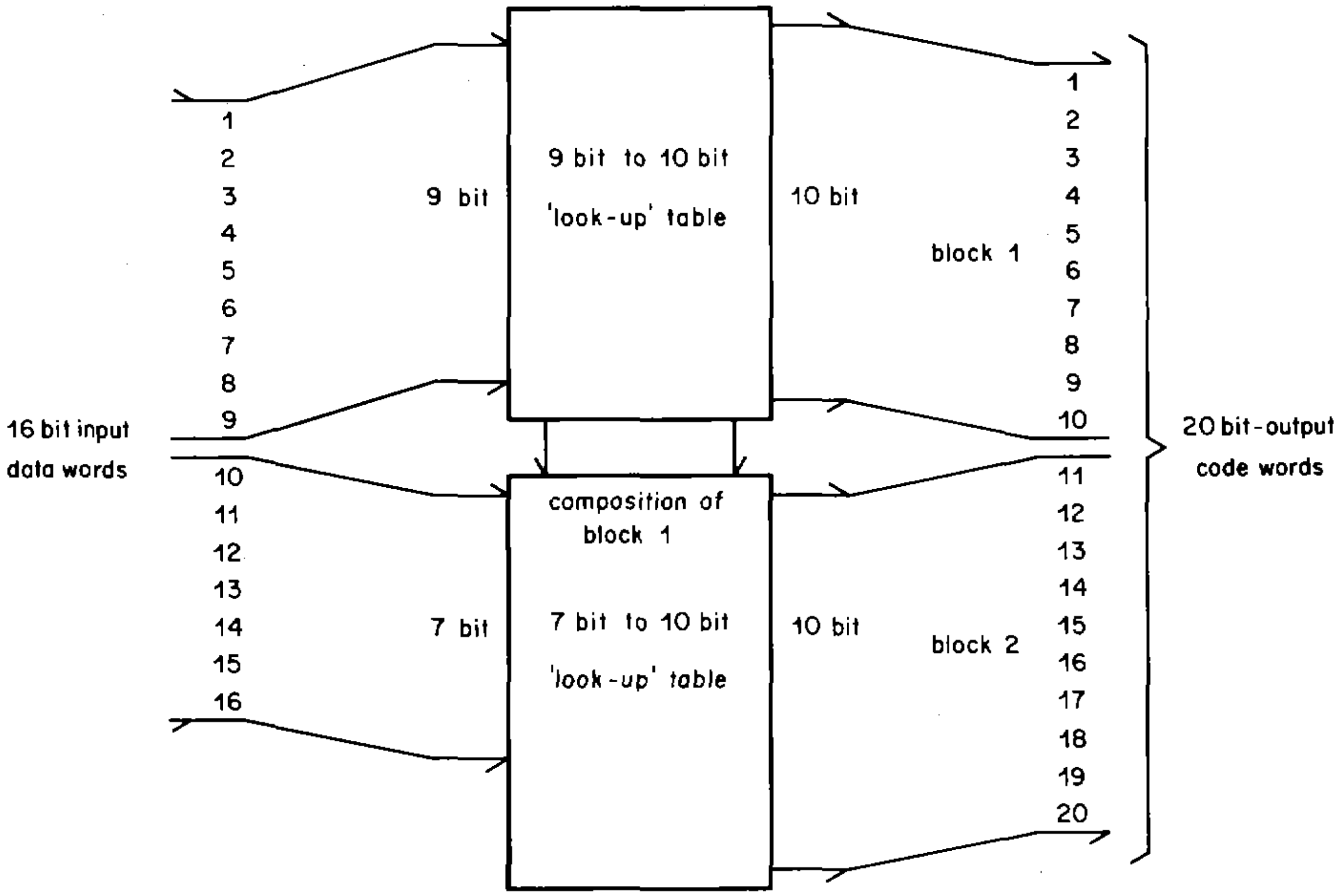

Fig. 1 — Schematic of a 20:16 coder.

same value. This limits the maximum string length of '1's and '0's at transitions between blocks, or between twenty bit words, to a maximum of five.

These conditions ensure that the serialised output stream has little magnitude at low frequencies. The data frequency spectrum, Fig. 2, shows that most components are contained in a four octave range.

On decoding the twenty bit words, the two ten bit blocks may be treated independently since each will uniquely define a particular data block value. Decoding is therefore achieved by applying the two ten bit blocks to look-up tables to reproduce the original data words.

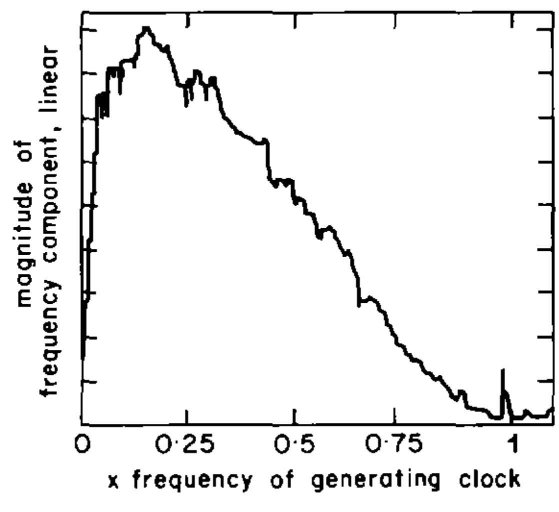

Fig. 2 — Spectral content of a scrambled video data stream coded with the 20:16 code.

3.2. Coding procedure

There are 512 ten bit output blocks which conform to the properties described in (a) and (b) of Section 2 above. This allows 9 bits of input data to be coded into the first ten bit block. This codeset may be divided into two equal subsets, each of 256 ten bit blocks, and each the ones complement of the other. The coding and decoding operations may therefore be simplified by connecting the first digit of the input 9 bit block directly to the first digit of the output ten bit code-block. The remaining eight input bits

are then connected to a PROM look-up table which outputs 9 bits of one subset of the output codeblocks. This output block is taken to an inverter acting on the nine output bits, the inverter being controlled by the first output bit. This is shown at the top of Fig. 3.

Having coded the first ten bits of output data as either a 4:6, 5:5 or 6:4 word, the second ten-bit block has been partially defined and must be either a 6:4, 5:5 or 4:6 word, respectively. The number of input bits that may be coded into the second output block is therefore limited, and will be the smallest number of ten-bit words which both conform to (a) and (b) in Section 2 above and have the property of being either 4:6, 5:5 or 6:4 words. There are 154, 4:6 words, 154 6:4 words and 164 5:5 words. This permits the coding of seven bits of input data into the second ten bits of output data.

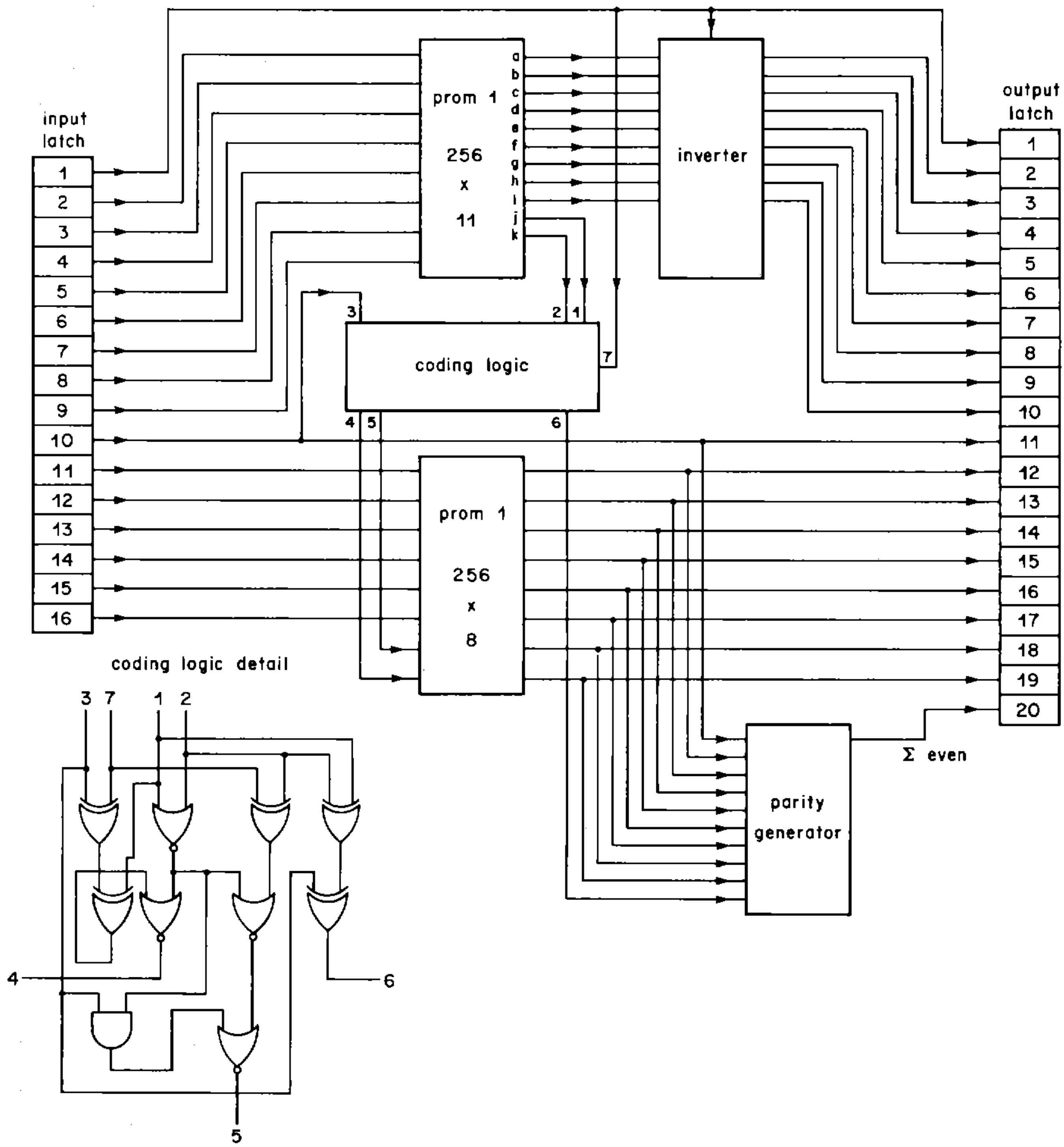

Fig. 3 – Implementation of the 20:16 coder.

To reduce the required PROM size, the first input digit of the second input block is connected to the first output digit in the manner described above, as shown in Fig. 3. Since all code words are constrained to have an equal number of '1's and '0's, the twentieth bit is defined after coding the first nineteen and is conveniently generated using parity generation circuitry, again, as shown in Fig. 3.

3.3. Decoding procedure

The decoding operation is considerably simpler than coding. The two ten bit blocks are taken to two 512 by 8-bit PROMs (Fig. 4). The first digit of block 1 is used to control an inverter operating on the input bits to a first PROM and, along with the first bit of the second input block, is fed directly to its position in the sixteen bit output word. The eight bit output of the first PROM and the six bit output of the second represent the remainder of the decoded 16 bit word.

The twenty bit words may be taken to parity checkers to test for any errors in the replayed data. If further error-detection is required, then additional PROMs can be used to check for invalid codewords.

4. Applications of the 20:16 code

The 20:16 code was devised initially for use with a helical scan Digital Video Tape Recorder (DVTR)[8] which recorded 7-bit quantised, reduced picture height, Y,U,V television signals sampled at 13.5 MHz, 6.75 MHz and 6.75 MHz respectively. The transport was modified C-Format machine with a six head cluster recording at a bit-rate of 40 Mbit/s per head (49 kbits/inch, 1900 bits/mm). Each head had an azimuth of several degrees with respect of the line of motion of the head, adjacent heads in the stack having opposite azimuth angles in order to reduce the required guardband between

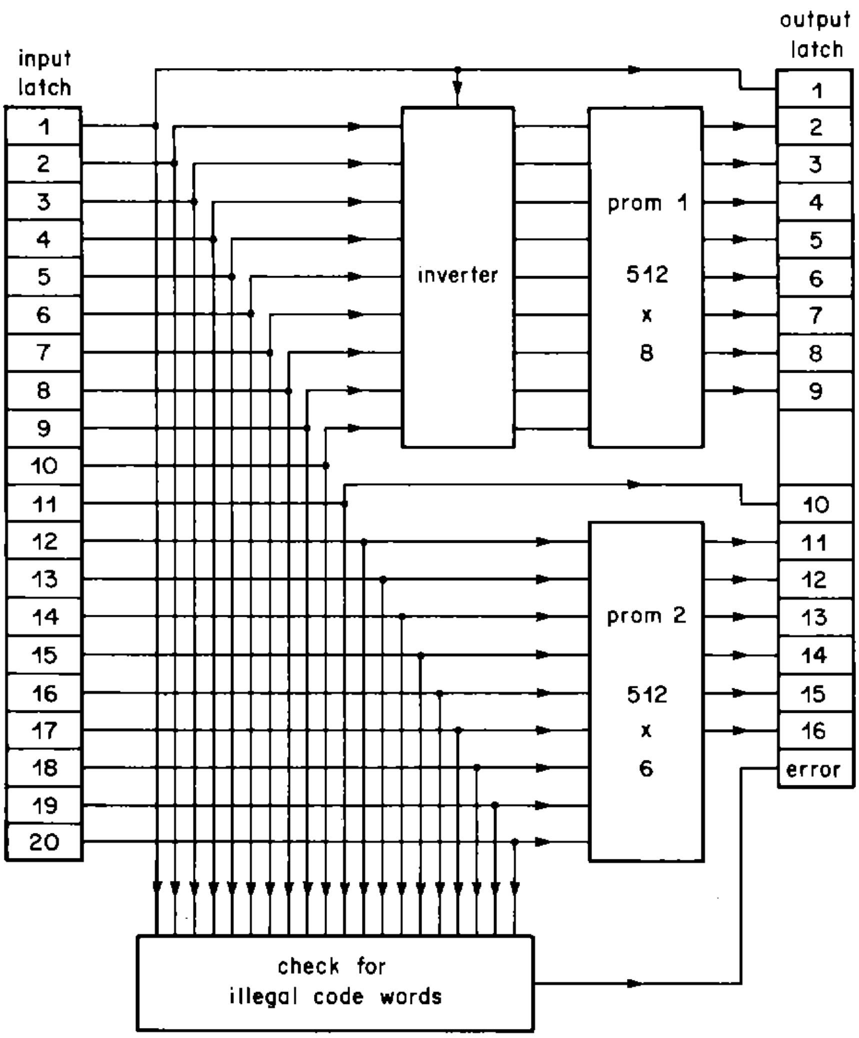

Fig. 4 – Implementation of the 20:16 decoder.

tracks. Fig. 5 shows the eye pattern of the replayed picture information from tape after equalisation. The off tape error rate was approximately 1 in 10^5, the majority of error events being dropouts; no correlation between picture content and error position was noticed.

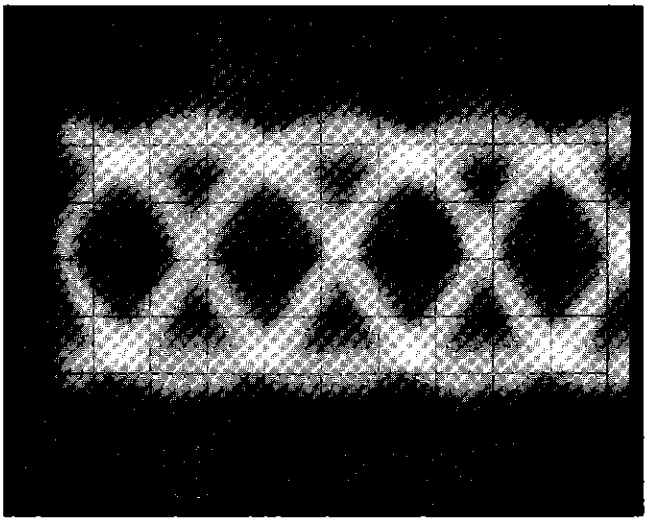

Fig. 5 — Eye Pattern at 40 Mbit/s of replayed data from a DVTR.

The code was also found suitable for serial video transmission and has been incorporated in an optical fibre link operating at 280 Mbit/s[9]. The link transmitted digitally encoded YUV television signals, according to Rec. 601 of the CCIR, up to a range of 1000 m.

5. Advantages of the 20:16 code

In a 20:16 coded serial data stream, a transition is guaranteed to occur at least every five bit-cells. The frequent occurrence of data transitions eases the demands on phase lock loops and results in accurate resampling of the recovered data. Crosstalk between adjacent tracks on the tape is also reduced. The depth of the recording into the tape is reduced allowing overwriting of data without the need separately erase the tape[1].

The code is capable of detecting all single bit errors and 93% of other error patterns. This proportion may be increased for burst errors by taking account of errors detected in adjacent words. Error detection information can contribute to the process of error correction to considerate advantage[10].

In assessing the performance of a channel code, the ability to regain correct word decoding phase from the serial data stream after a disturbance is of importance. An advantage of the 20:16 structure is the existence of 20 bit synchronising words which conform to the properties of the code, but which are distinguishable in a sequence of serialised 20:16 codewords. This is because particular bit patterns may be chosen for the synchronising words which do not occur in the serialised data stream, regardless of the sequence of input data values being coded or the phase of the serial to parallel conversion relative to the bit stream (decoding phase). These words may be regarded as lying outside the 16 bit codeset of the 20:16 code. There are forty-eight suitable words and these are listed in the Appendix (Section 8).

A simple framing system may be devised whereby incoming serial data is examined for the bit pattern corresponding to a framing word for each decodingphase. When this pattern is recognised the correct incoming word phase is known. The frequency of incorrect framing during periods of interference is inherently low since the chance of finding a framing word in a random data stream is about 1 in 2^{20} or 1 in 10^6 for each decoding phase. This high level of security allows the same framing words to be used for other purposes such as for television line and field synchronisation signals. It also avoids the need to use complex maximum probability algorithms, avoids the need to assign input data codes for framing purposes and hence reduces synchronising overheads.

6. Conclusions

A new channel code has been described which offers several advantages over contemporary codes in terms of its spectrum, clocking and framing information, and error detection. These not only improve channel performance but can also simplify instrumentation. The code has been used successfully both for high density digital magnetic recording on a helical scan transport and for high bit-rate optical fibre transmission.

7. References

1. DOLBY, D., LEMOINE, M., FELIX, M. Formats for digital video tape recorders, International Broadcasting Convention 1980, Conference Publication 191 Page 231.

2. JACOBY, G.V. 'A New Look-Ahead Code for Increasing Data Density' IEEE Trans. MAG-13, No. 5 pp 1202-1204, September 1977.

3. BALDWIN, J.L.E., BELLIS, R.A. *and* IVE, J.E.S. UK Patent Application No. GB 2001789A.

4. Philips Technical Review Vol. 40 1982 No. 6.

5. JONES, A.H. *and* CHAMBERS, J.P. 1982. 'Digital magnetic recording: conventional saturation techniques' BBC Research Department Report No. 1972/9.

6. Proceedings Letters, Proceedings to IEEE. July 1969. p 1314.

7. PARKER, M.A. DVTR 20:16 codec, UK Patent Application No. 8127791.

8. PARKER, M.A. *and* BELLIS, F.A. 1982 'Signal processing for an experimental digital video tape recorder'. Fourth International Conference on Video and Data Recording, Southampton, 1982. IERE, London, Conference Proceedings No. 54, pp. 207-215.

9. MARSDEN, R. 1983. 'Optical fibre links for digital YUV signals' BBC Research Department Report No. RD 1983/14.

10. PARKER, M.A. 1983. 'A range of combined error correction and record code schemes'. BBC Research Department Report No. RD 1983/16.

8. Appendix

20:16 Framing Words

The full range of 20 bit words that can be used for framing are given below. Each 20 bit word is defined in the list as a group of five hexadecimal characters.

2BEA1	33DC1	33EC1	387D8	3C13E	3C23E
3E83C	437CC	43C3E	457D4	467D8	67682
75057	7827C	7847C	78783	79067	79827
79847	7A82B	7C3C1	7C782	7C833	7D078
82F87	837CC	8387D	83C3E	857D4	86768
867D8	86F98	8787C	87B83	87D83	8AFA8
9847D	B9827	BA82B	BC3C1	BC833	C17C3
C1C3E	C3DC1	C3EC1	C7827	CC13E	CC23E

Printed by BBC RESEARCH DEPARTMENT, Kingswood Warren, Tadworth, Surrey, KT20 6NP